Facultad de Informática
Universidad Complutense de Madrid

ANUNCIO DE CONFERENCIA

Specular Effects on the GPU: State of the Art.

Dr. Gustavo Patow

Grupo de Geometría y Gráficos. Universitat de Girona

Sala de Grados • 23 de abril de 2009 • 18: 00
entrada libre hasta completar el aforo

resumen:

Mirrors, very shiny metallic objects, refracting transparent surfaces, such as glass or water, significantly contribute to the realism and the beauty of images. Thus, their simulation in virtual environments has been an important task since the early days of computer graphics. Unfortunately, such phenomena do not fit into the local illumination model of computer graphics, which evaluates only the direct illumination, i.e. light paths originating at a light source and arriving at the eye via a single reflection. In a mirror, however, we see the reflection of some other surface illuminated by the light sources, thus rendering a mirror requires the simulation of longer light paths responsible for the indirect illumination. While in the local illumination model a surface point can be shaded independently of other surface points in the scene, indirect illumination introduces a coupling between surfaces, so the shading of a point requires information about the scene globally.

Before 2006, GPUs could only be controlled through graphics APIs such as Direct3D or OpenGL. These APIs are based on the concepts of the incremental rendering pipeline and present the GPU to the application developer as a pipeline of programmable vertex, geometry, and fragment shaders, and also non-programmable but controllable fixed function stages. One of these fixed function stages executes rasterization, which is equivalent to tracing a bundle of rays sharing the same origin, and passing through the pixels of the viewport. The first hits of these rays are identified by the z-buffer hardware. Since this architecture strongly reflects the concepts of incremental rendering, and processes a vertex and a fragment independently of other vertices and fragments, the utilization of the GPU for other algorithms requires their translation to a series of such rendering passes. The result of a pass can be stored in a texture that can be read by the shaders during the subsequent passes.

The objective of this presentation is to review algorithms that can render specular, i.e. mirror reflections, refractions, and caustics on the GPU. We will establish a taxonomy of methods based on the three main different ways of representing the scene and computing ray intersections with the aid of the GPU, including ray tracing in the original geometry, ray tracing in the sampled geometry, and geometry transformation. Having discussed the possibilities of implementing ray tracing, we will consider the generation of single reflections/refractions, inter-object multiple reflections/refractions, and the general case which also includes self reflections or refractions. Moving the focus from the eye to the light sources, caustic effect generation approaches will also be examined.